KABARAK UNIVERSITY

SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DESIGN OF A HEAVY HAULAGE VEHICLE'S LOAD MANAGEMENT SYSTEM

BY

MIKE MWITI

SUPERVISED BY: DR. CHRISTOPHER MAGHANGA

A RESEARCH PROJECT SUBMITTED TO THE DEPARTMENT OF TELECOMMUNICATIONS IN THE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY IN PARTIAL FULLFILMENT FOR THE AWARD OF A DEGREE IN BACHELOR OF SCIENCE IN TELECOMMUNICATION.

DATE: 28TH APRIL 2022

Declaration

This research project is my original work and has not been presented for a degree at any other university.

Name	Reg no.	Sign	date
Mike Mwiti	TLCM/MG/1425/05/18	……	……..

SUPERVISOR'S DECLARATION

This project has been submitted for examination with my approval as the university supervisor

Supervisor's name: DR. CHRISTOPHER MAGHANGA

Signature …… Date……...

DEDICATION

I dedicate this project to: My Supervisor who has given me the needed guidance to carry out this project, my fellow classmates for the necessary support they gave me when I needed it and most of all to my family whom have given me both financial and moral support.

ABSTRACT

Maintaining road pavement quality in several areas in Kenya is essential because most goods are distributed mainly by land transportation. Road transport carries over 93% of all goods and passenger traffic in the country. Having an adequately developed and well maintained roads network improves road transport; reduces operating costs (business, fuel and spare parts); it also supports growth in other sectors such as tourism, agriculture and industrial sector with consequent increase in employment and income opportunities; road safety; and improves the socio-economic wellbeing of a nation by stimulating overall economic growth in both domestic and international trade and facilitating easy flow and access to manufactured goods. The Northern Corridor is one of the busy transport routes in East and Central Africa. It provides a lifeline route through Kenya to the landlocked countries. An extensive network of transport routes originates from the Port of Mombasa, through Uganda, then branching off to Rwanda, Burundi and the eastern parts of DRC. Therefore, efficiency in the performance and operations at Gilgil weighbridge station is critical to the enhancement of cargo movement along the corridor and overall trade among the East African countries, especially Kenya, Uganda and Rwanda.

Based on this issue, a load management system was developed. The main strategy of this load management system is to be able to measure the total weight of the vehicles by measuring the weight exerted by the load on the axles. This method enables the measurement of the vehicle weight to be carried out while the vehicle engine is still on or off, and therefore if the vehicle is overloaded the engine won't start, or stop if it was already running. Then if the vehicle is overloaded the authorities at the weighbridges are informed via SMS (short message service). Therefore, all the heavy vehicles integrated with the heavy haulage load management system can easily be evaluated by the weighbridge authorities thus saving a lot of time and increasing weighbridge efficiency.

In this research project, a simple device with a weight measurement system is developed. As a case study, the prototype is implemented using a loadcell that can be mounted on the axle and a gear motor to simulate the vehicle's engine. The system prototype is based on a load sensor for the weight measurement and Arduino microcontroller with the addition of alarm and LED connection and the alert system is implemented using a GSM module.

TABLE OF CONTENTS

CHAPTER ONE
INTRODUCTION
1.0 Background

Maintaining road pavement quality in several areas in Kenya is essential since most goods are distributed mainly by roads (Anyango, 2011). Gilgil weighbridge is one of the five Axle Load Control Stations located along the Northern Corridor road in Kenya. Mariakani and Athi River weighbridge stations which were usually busier than Gilgil weighbridge station were redesigned and new constructions were on-going. The facilities at the station included three weighbridges, holding bays, offices, toilets, public display units, digital readers, screening lane, CCTV Cameras, entrances and exits. The main operations at the weighbridge involved weighing of heavy commercial vehicles with at least 7 tons of cargo. Those heavy haulage trucks with loads exceeding the allowed legal limit were stopped from proceeding by Kenya Police and corrective measures enforced in accordance with the Traffic (Amendment) Act 2013. Prior to 1978/1979 there had been good enforcement of load limits. These had been relaxed with almost no control from 1979 in response to an appeal from Rwanda, where the legal limits had been much higher. In 1984 the enforcement of the load limits was re-introduced, but again in response to an appeal from inland countries, this was relaxed.

Then in 1997 El Nino rains had triggered the almost total collapse of the Northern Corridor Road (European Union, 2006). In 1998 the Government of Kenya took action to strengthen the enforcement of existing laws to control axle-loads. This became necessary since a huge proportion of heavy haulage trucks plying the Northern Corridor were grossly overloaded. Furthermore, it was realized that roads with a theoretical design life span of fifteen years were failing after less than five years.

The enforcement of existing laws on axle load limits now constitutes a firm commitment of the Government of Kenya to establish a sound-operating framework for the rehabilitation of the deteriorated transport infrastructure. In addition, the Government agreed to vigorously enforce axle load limits as part of the accompanying measures established with the European Commission for support to the road sector in general and to the rehabilitation of the Sultan Hamud – Mtito Andei and Mai Mahiu – Naivasha – Lanet roads (European Union, 2006).

Assessment of the Operations of Weighbridges in Kenya: Case of Gilgil Weighbridge 2 September 2016 the Government further agreed that an independent monitoring of axle load enforcement should be carried out. The objective of independent monitoring was to determine the enforcement and effectiveness of existing measures to control axle loads and to make recommendations to ensure greater conformity and compliance. The exercise was carried out by a consultant (Otieno Odongo & Partners) between February 1999 and March 2001 with random monthly visits to the static weighbridge stations at Mariakani, Athi River, Gilgil and Webuye (European Union, 2006).

The Ministry of Roads' Road Sector Investment Programme (2010-2014) report estimated a maintenance backlog for the paved network at Kshs. 230 billion. In addition, there was an annual

maintenance cost of Kshs. 40 billion as at 2014. These requirements against approximately Kshs. 24 billion available for road maintenance indicate the inadequacy of available funds. This project aims to eliminate overloading on the roads, thereby reducing maintenance cost burden on the government. According to the Ministry of Transport's National Integrated Transport Policy (2009), road transport carries over 93% of all good and passenger traffic in the country. Having an adequately developed and well maintained roads network improves road transport; reduces operating costs (business, fuel and spare parts); supports growth in other sectors such as tourism, agriculture and industrial sector with consequent increase in employment and income opportunities; road safety; and improves the socio-economic wellbeing of a nation by stimulating overall economic growth in both domestic and international trade and facilitating easy flow and access to manufactured goods.

The Northern Corridor is one of the busiest and most important transport routes in East and Central Africa. It provides a lifeline through Kenya to the landlocked countries like Uganda, Rwanda, Burundi and DR Congo. An extensive network of transport routes originates from the Port of Mombasa, through Uganda, then branching off to Rwanda, Burundi and the eastern parts of DRC, with the largest mode being road transport (East African Online Transport Agency, 2012). Therefore, efficiency in the performance and operations at Gilgil weighbridge station is critical to the enhancement of cargo movement along the corridor and overall trade among the East African countries, especially Kenya, Uganda and Rwanda Based on this issue, a load management system was developed.

The main strategy of this system is to measure the total weight of the vehicles by measuring the weight exerted by the load on the axles. This method enables the load weight measurement to be done while the vehicle engine is still on, and therefore if the vehicle weight of the vehicle exceeds the required limit, the engine won't start, or stop if it was already running. Then if the system is overloaded the authorities at the weighbridges are informed via sms. Therefore, all the heavy vehicles integerated with the heavy haulage load management system can easily be evaluated by the weighbridge authorities thus saving a lot of time and increasing weighbridge efficiency.

In this project report, a simple device with load weight detection system is developed. As a case study, the prototype is implemented using a loadcell that can be mounted on the axle and a gear motor to simulate the vehicle engine. The system prototype is based on a load sensor, while the load weight measurement system is developed using Arduino microcontroller with an addition of an alarm and LED connection and the alert system is implemented using a GSM module.

1.1 Statement of The Problem

Control of overloading through setting up of weighbridges is intended to eliminate the huge damaging effects of overloaded vehicles on pavements. However, a joint study by JICA and PADECO (2008) stressed that the effectiveness of any weighbridge station is dependent on the type of weighbridges provided, the accompanying facilities, management set up and location among other factors. Chan (2008) noted in his study that sometimes weighbridges have just been provided without careful consideration of their adequacy and the requirements. Therefore, if this load management system is adopted to the already existing weighbride platform, the whole system as whole will be more effective, and the necessary laws will be enforced efficiently, thus solving the following problems:

Inadequate capacity at the weighbridge station to handle the traffic at the station as evidenced by long queues at the weigh bridge stations. The delays mean it takes longer to transport cargo along the corridor and this translates to higher transport costs. The congestion also affects the free flow of other vehicles passing through the weighbridge area especially where the queues extend beyond the start of the screening lane.

The existing weighbridges at the station only weigh one axle at a time, which results into longer service times per vehicle. This further contributes to the long delays experienced at the station. This project can solve this since it measures the weight of all the axles at once, before it even arrives at the station.

Kenya National Highways Authority (KeNHA) is responsible for the Gilgil Weighbridge Station, is unable to monitor in real time the weighbridge station, where they have contracted a Management Contractor to manage the operations and supervise improvements. They rely on data relayed later and this possibly contributes to some level of reduced transparency and accountability and creates room for manipulation of data to satisfy the requirements and targets set by KeNHA. This system easily solves this because data can be received from the vehicle itself in real time.

1.2 General Objective

To design a system that can weigh vehicle load weight, detect if its overloaded, send sms alert and stop the vehicle's engine.

1.2.1 Specific Objectives

I. To identify a system that can weigh vehicle load weight.
II. To incorporate a system that can send a message alert if the vehicle is overloaded.
III. To incorporate a system that can stop the engine if the vehicle is overloaded.

1.3 Relevance to telecommunications

This project will utilize a GSM (Global System for Mobile Communication) module which is a key component in telecommunications. This module is used to connect the system to mobile communication resources. Thus, the system will utilize the telecommunications infrastructure and resources to constantly send real time data. Therefore, this becomes relevant to telecommunications since it enables the telecommunications sector to grow and be utilized in the transportation sector in return increasing the overall revenue of the telecommunication industry.

CHAPTER TWO
LITERATURE REVIEW
2.0 Introduction

The requirement to weigh civilian traffic vehicles is a thought that dates back to 1741 when the UK government of the day introduced the Turnpike act, which decreed that toll was to be paid for use of roads according to the weight of the vehicle. Massive steelyards were installed, but vehicles had to be lifted before their weight could be measured. The solution to this lay in the production of platform scales or weighbridges onto the roads. From late 1940's mechanical weighing began to combine with electronics, but it was not until the invention of the load cell that complex and bulky systems and knife edges were replaced.

The two main types of static weighing systems in use today consist of stationery platform and portable wheel scales. The accuracy of both systems makes it eligible for enforcement purposes. A truck scale consists of a scale frame that supports the weight of a truck without major bending, a number of load cells, junction boxes and weight indicator. These traditional platform scales are available in a wide range of sizes and weighing capacities, in both pit mounted and surface mounted versions which calculate to an accuracy of less than 0.5%. Portable wheel load scales have been developed to allow for measuring wheel and axle loads. Each wheel is measured individually, although their precision is somewhat lower than platform scales. Depending on how many scales are used, additional errors may be introduced because of weight transfer between the axles due to longitudinal tilting of the vehicle, incorrect sensor levelling, Site unevenness, sensor tilting, mechanical friction in the suspension and friction forces induced by braking.

The influence of these factors on results, is reduced by using the same number of scales as number of wheels in the axle group or the whole vehicle. A set of 6 wheels load scales can achieve a maximum error band of less than, but they are slow and require a lot of labor. A set of 2-wheel load scales can achieve a maximum error band between 1% (good site and vehicle in good condition) and 35% (average site and vehicle in poor condition) for the GVW (GROSS VEHICLE WEIGHT). Invented by the Romans in 200BC, steelyards consisted of a beam with a sliding poise to counterbalance the load. 14 Static scales offer advantage of allowing accurate calculation of the vehicle weight. However, from the data collection and weight enforcement perspective, they are subject to a number of drawbacks.

Benekohal et al (1999) conducted a study at a static weigh station in Illinois where it was found that 30% of all trucks could not be weighed because the weigh station was temporarily closed to prevent a queue. In addition, in average a truck was delayed by approximately 5 minutes. Aside from the inconvenience imposed on truck drivers, a greater problem, exists with regard to avoidance of weight enforcement stations by overweight trucks. Cunagin et al (1997) showed that the number of overweight vehicles decreases with increased enforcement activity, however vehicles attempt to bypass these permanent truck weight enforcement stations. It was found that violations at permanent weight enforcement stations were minor, whereas those on the bypass routes were much more severe. A total of 0.8% of the trucks were overweight at the fixed scales, whereas 19% were in violation on the bypass routes during the study.

Recently WIM measuring campaigns in Sweden have shown that the level of overloading cannot be estimated from static weighing, due to the avoidance of Police weighing locations by offending drivers. It is believed that truck operators would continue to operate overweight vehicles as long as they can gain an economic advantage by either evading or paying a fine less than the profit received from overloading (Cunagin, 1997). This project offers a solution to this problem by allowing vehicles to be weighed as they travel but especially when they are being loaded, as part of integrated system where trucks can be pre-selected via the system prior to entering the weighbridge station.

GSM Architecture

There are four major components of the GSM network architecture.

I. **Base Station Subsystem**

The Base Station Subsystem (BSS) includes the Base Transceiver Station (BTS), which is responsible for providing connectivity within cell in which its located, and the Base Station Controller (BSC), which takes care of switching between multiple BTS.

II. **Network Switching Subsystem**

It is also known as the Core network, the Network Switching Subsystem (NSS) consists of the Mobile Switching Centre and its associated components. Its main responsibility is to switch between multiple channels and to ensure mobility management (tracking subscribers). They're generally owned and managed by the service providers themselves and further connect to an interconnected network known as the Public Switched Telephone Network (PSTN).

III. **GPRS core network**

The GPRS core network is the central part of the general packet radio service (GPRS), which allows GSM mobile networks to transmit data over the internet. It provides mobility management, session management, and transport for the IP packets (that contain the data, in the form of images, videos, or text, to be transmitted over the internet).

This network is also responsible for billing of subscribers by keeping a record of their data, call history, and usage.

IV. Operations Support System (OSS)

The Operations Support System (OSS) is used by service providers to take care of their cellular network. It mainly does the following functions:

- Network Inventory – keeping track of all the components in the network.
- Service Provisioning – preparing a network to allow it to provide a new service for the customers.
- Network Configuration – this is process of ensuring that the performance and functionalities of the components are never compromised.
- Fault Management – a set of operations that detect, isolate, and manage malfunctions in the network.

2.1 Review Of Similar Works

Similar systems created in the area of managing and monitoring heavy haulage vehicles include the following.

I. Design and Prototyping of Weigh-In-Motion and Overload Detection System for Freight Vehicle.

The high number of freight vehicles that exercised overloading in Indonesia led to a significant reduction in the quality of road pavement(A H Masyhur et al 2019). The overload detection system using weigh bridges have been ineffective, since they require vehicles to maneuver and enter the weighing station. In the system provided, after manual checking, overweight vehicles are penalized. The Weigh-in-Motion (WIM) device that is developed in this research utilized a bending plate sensor with strain gage. As an overloaded vehicle run over the WIM, axle weight would be automatically recorded, and the data is processed further to assess the compliance to maximum freight weight as specified for the vehicle. In this research work, the WIM prototype is developed and tested using a motorcycle under normal operation and when overloaded at different speeds. Arduino based signal processing and analysis system has been developed such that the overload detection may be performed automatically. When the system senses an overloaded vehicle, it will automatically trigger an alarm and a camera will capture the vehicle. The errors of the prototype is within acceptable range and the system has shown the potential to detect whether the vehicle, i.e., the motorcycle, is overloaded or not

II. The Smart Vehicle Management System for Accident Prevention by Using Drowsiness, Alcohol, and Overload Detection.

The principal intent behind this project is to make a smart vehicle management system that will prompt us to reduce traffic accidents and damage of road infrastructure (UNIVERSITY OF CONNECTICUT 2021). There are three major elements in this prototype of the smart vehicle management system. First, there is a drowsiness detector, which automatically identifies the drowsiness of the driver throughout driving time. Secondly, an alcohol detector will trace if there is any presence of alcohol on the body of that particular driver. Lastly, there is an overload detector, which will show if the vehicle is overloaded, or not.

2.2 Theory on the system circuit blocks.

I. ARDUINO UNO R3

The Arduino Uno is a microcontroller board based on the ATmega328. It has 20 digital input/output pins; 6 can be used as power outputs and 6 can be used as analog inputs and the rest for digital output, a 16 MHz resonator, a USB connection, a power jack, an (ICSP) in-circuit system programming header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC to DC converter or battery to get started.

The Arduino uno r3 will be utilized to integrate all the components and upload the code used to execute the whole system.

FIGURE 1. Arduino Uno R3

II. GSM MODULE

The SIM800L is a cellular communication module that can make calls, send email and also send SMS, and even connect to the internet. The module is intended to operate like a mobile phone, but it requires external peripherals to function properly. The SIM800L has a lot of functions but in this project only the SMS function was utilized.

The SIM800L needs to be powered by 3.7V which is a common standard among most cellular modules. One might think of using the Arduino's 5V and 3.3V supply from the Arduino but this is not applicable. The SIM800 is specified to use a power supply in the range of 3.4V to 4.3V. So, using 5V could damage the GSM module and 3.3V is not enough to reliably power it. Therefore, an external 3.7V Li-ion Polymer battery or charger can be used as the power source for the module.

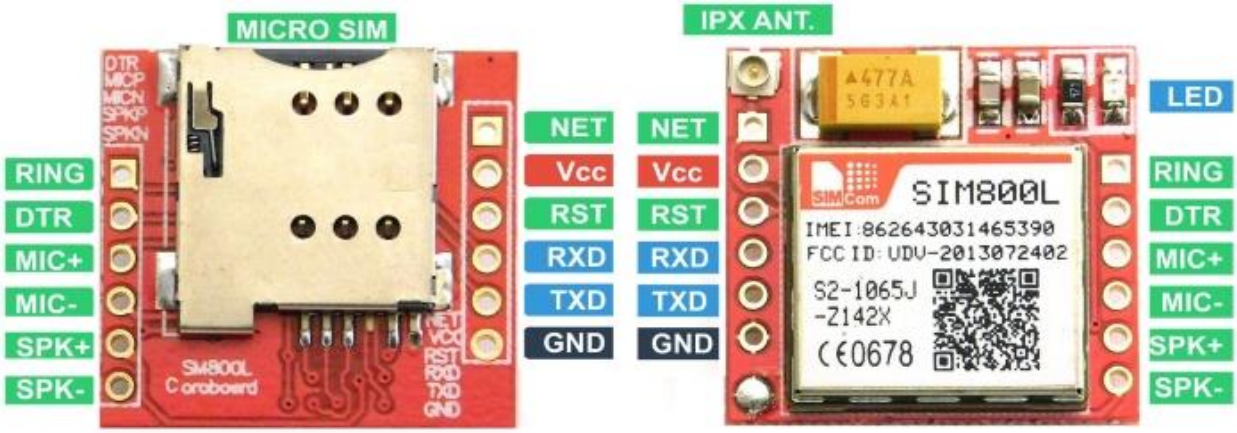


FIGURE 2. GSM SIM800L module

III. LOAD CELL

A load cell is made by using an elastic member (with very highly repeatable deflection pattern) to which a number of strain gauges are mounted. When the load is applied to the body of a resistive load cell, the elastic member, deflects and creates a strain at those locations due to the stress applied. As a result, two of the strain gauges are in compression, whereas the other two are in tension.

The four strain gauges are configured using a Wheatstone Bridge configuration with four separate resistors connected as shown in figure 3, which is called a Wheatstone Bridge Network.

An excitation voltage of 10V is applied to one set of corners and the voltage difference is measured between the other two corners. At equilibrium with no applied load, the voltage output is zero or very close to zero when the four resistors are closely matched in value. That is why it is referred to as a balanced bridge circuit.

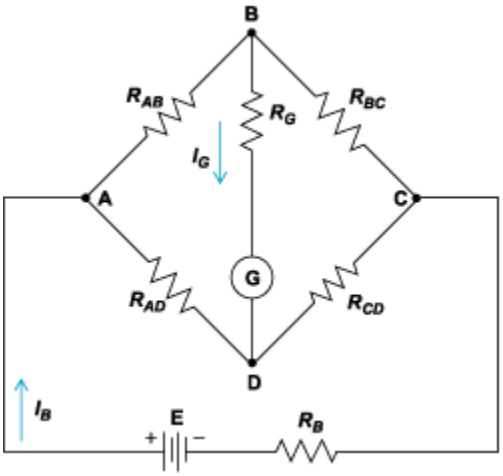


FIGURE 3. Wheatstone Bridge

When the metallic member to which the strain gauges are attached, is stressed by the application of a force, the resulting strain – leads to a change in resistance in one (or more) of the resistors. This change in resistance results in a change in output voltage. This small change in output voltage usually about 20 mV can be measured and digitized after careful amplification of the small milli-volt level signals to a higher amplitude 0-5V or 0-10V signal.

IV. HX711

This module uses a 24 high-precision A / D converter. ThE chip is designed for high-precision electronic scale and design, it has two analog input channels and a programmable gain of 128 integrated amplifier. The input circuit can be configured to provide a bridge voltage electrical bridge such as pressure or load sensor model.

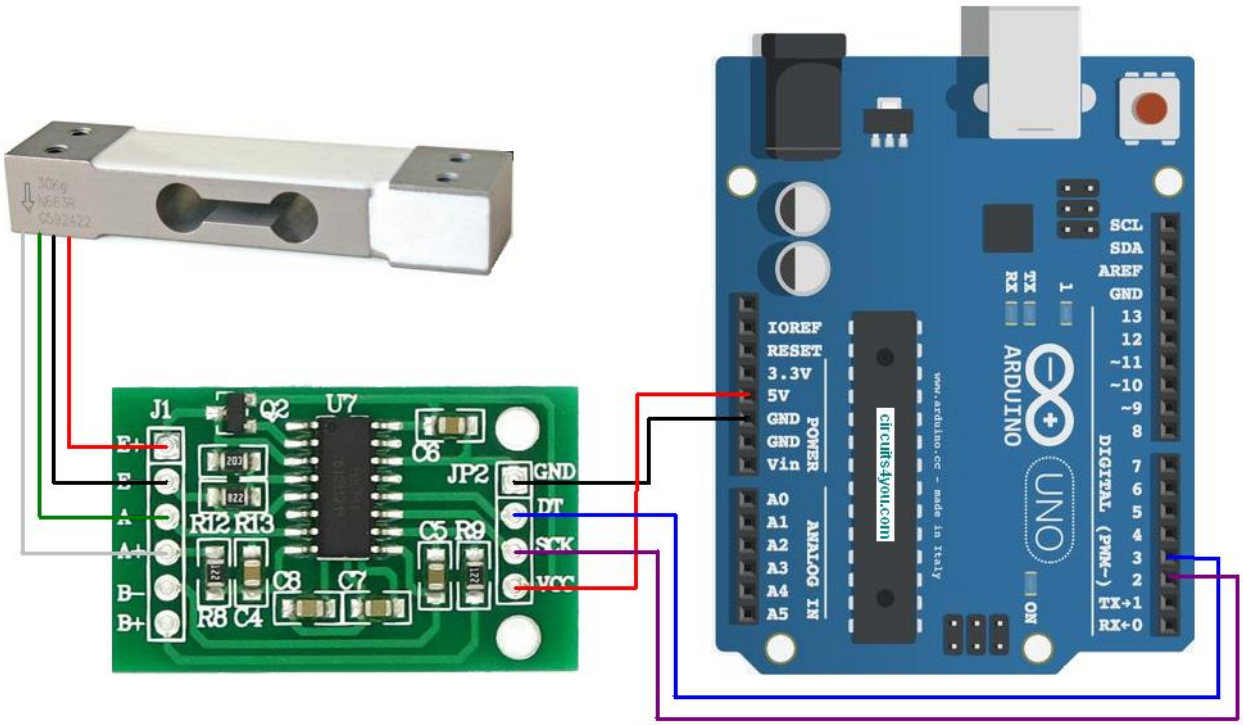


FIGURE 4. HX711 load cell amplifier.

V. ALARM SYSTEM (LED AND BUZZER)

A buzzer or beeper is an audio signaling device, which can be mechanical, electromechanical, or piezoelectric. Most applications of buzzers and beepers include alarm devices, timers, and confirmation of user input for example, mouse click or keystroke.

A piezoelectric speaker (in short “piezo buzzer”,) is a loudspeaker that uses the piezoelectric effect to produce sound. The mechanical motion is created by applying a voltage to the piezoelectric material, and this motion is converted to audible sound using diaphragms and resonators.

Compared to other speaker designs piezoelectric speakers are easier to drive; for example, they can be connected directly to TTL (Transistor-Transistor Logic) outputs, although more complex drivers can give greater sound intensity. Typically, they operate well in the range of 1-5 kHz and up to 100 kHz in ultrasound applications.

Piezoelectricity, is the electric charge that accumulates in certain solid materials such as crystals, certain ceramics, and biological matter such as bone, in response to applied mechanical stress. The word piezoelectricity means electricity resulting from pressure and latent heat.

FIGURE 5. Piezo Buzzer

How LED works

LEDs are the most capable maens to turn electric current into illumination. When a current flows through a diode in a forward direction, it contains surplus electrons that move in one direction in the lattice and “holes” (voids in the lattice) moving in the other. Occasionally, electrons can recombine with holes. When they do, the process releases energy in the form of photons.

This is true of all semiconductor junctions, but LEDs use materials that maximize the effect. The color of the light emitted (corresponding to the energy of the photons) is determined by the semiconductor materials which form the diode junction.

Appearing as practical electronic components in 1962, the earliest LEDs emitted low-intensity infrared (IR) light. Infrared LEDs are used in remote-control circuits. Recent developments have produced LEDs available in visible, ultraviolet (UV), and infrared wavelengths, with high and low, or intermediate light output. For example, white LEDs are suitable for room and outdoor lighting. LEDs have also given rise to new types of displays and sensors, while their high switching rates are useful in advanced communications technology with applications as diverse as aviation lighting, fairy lights, automotive headlamps, advertising, general lighting, traffic signals, camera flashes, lighted wallpaper, and medical devices.

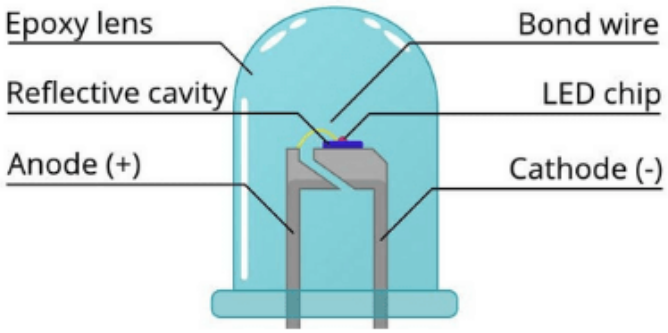


FIGURE 6. LED

VI. ENGINE/ ELECTRIC MOTOR

An electric motor is an electrical machine that converts electrical energy into mechanical energy. Most electric motors operate by interaction between the motor's magnetic field and electric current in a wire winding in order generate force in the form of torque, which is applied on the motor's shaft. An electric generator is mechanically identical to an electric motor, but it operates with a reversed flow of power, converting mechanical energy into electrical energy. Electric motors can be powered by direct current (DC) sources, such as from batteries, or rectifiers, or by alternating current (AC) sources, such as a power grid, inverters or electrical generators.

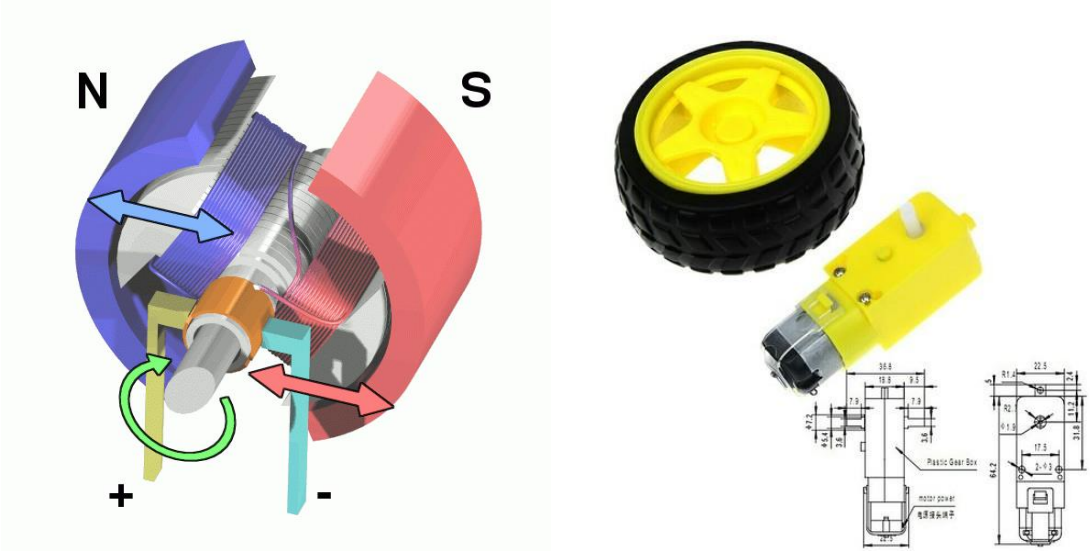


FIGURE 7. Motor.

CHAPTER THREE
METHODOLOGY
3.0 Description Of The Methods.
3.0.1 To identify a system that can weigh vehicle load weight.

The first objective of this project was to create a system that can weigh the load of a vehicle. This was achieved by the use of a Load Cell, HX711 load cell amplifier and Arduino uno R3.

A load cell is made by using an elastic member to which a number of strain gauges are attached. When the load is applied to the body of a resistive load cell, the elastic member deflects and creates a strain at those locations due to the stress applied. As a result, two of the strain gauges are in compression, whereas the other two are in tension. The HX711 module is utilized to amplify the load cell, it uses 24 high-precision A / D converter. This chip is designed for high-precision electronic scale and design, has two analog input channels. The input circuit can be configured to provide a bridge voltage electrical bridge, sensor model. it's an ideal high-precision front-end module.

Arduino Uno R3 is a microcontroller, its used to calibrate the Load cell using the amplifier the amplifier ports SCK (clock signal) and DOUT (data out) are connected to the digital ports of the microcontroller. The VCC (voltage supply) and GND (ground) ports are connected to the consecutive ports of the microcontroller. The SCK and DOUT ports are used by the microcontroller to receive and send signals. Therefore, when the calibration was done the load cell was able to read the weight of up to one kilogram which can be used to simulate the vehicle axle weight. The basic circuit is as shown below;

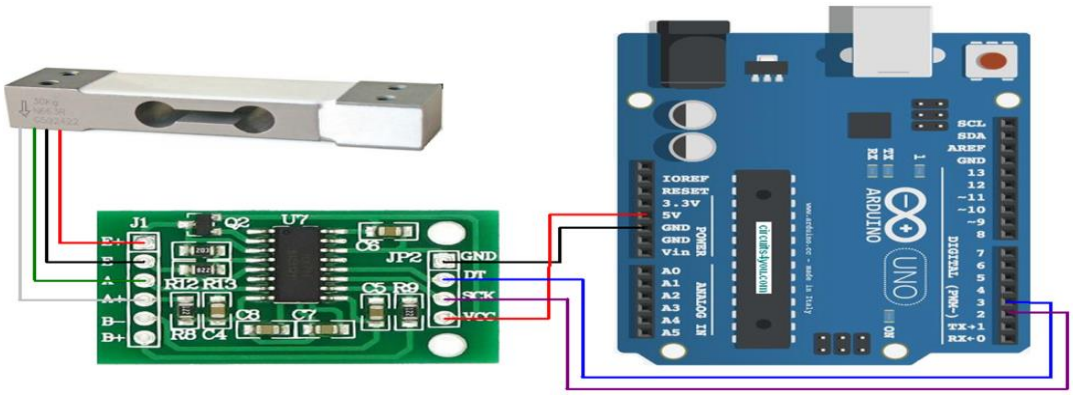


FIGURE 8. LOAD CELL/ HX711 CIRCUIT

3.0.2 To incorporate a system that can send a message alert if the vehicle is overloaded.

The second objective was to make a system that can send a message alert to the authorities when the vehicle is overloaded. This objective was achieved by the use of a GSM module. The GSM module has serial ports on both sides of the circuit, the left side is meant for power supply, network, reset, receiving and sending sms. The right side is meant for making calls, therefore for this project only the left side in the figure shown below is utilised.

FIGURE 9. GSM MODULE

The network port is used to attach an antenna, the VCC and GND ports are connected to the corresponding ports on the microcontroller, then the RXD and TXD ports are connected to the serial TXD and RXD digital ports of the microcontroller respectively. When the sim card is inserted to the module it has to have airtime, when inserted the module searches for the network and it blinks once every 3 seconds when the connection is established.

In the GSM code the number to which the SMS will be sent is included and then the code is integrated to the HX711 code so that when the vehicle is overloaded a message is sent.

3.0.3 To incorporate a system that can stop the engine if the vehicle is overloaded.

The final objective was to make sure the vehicle engine stops when the vehicle is overloaded. this was achieved by connecting the DC electric motor as shown in the figures below.

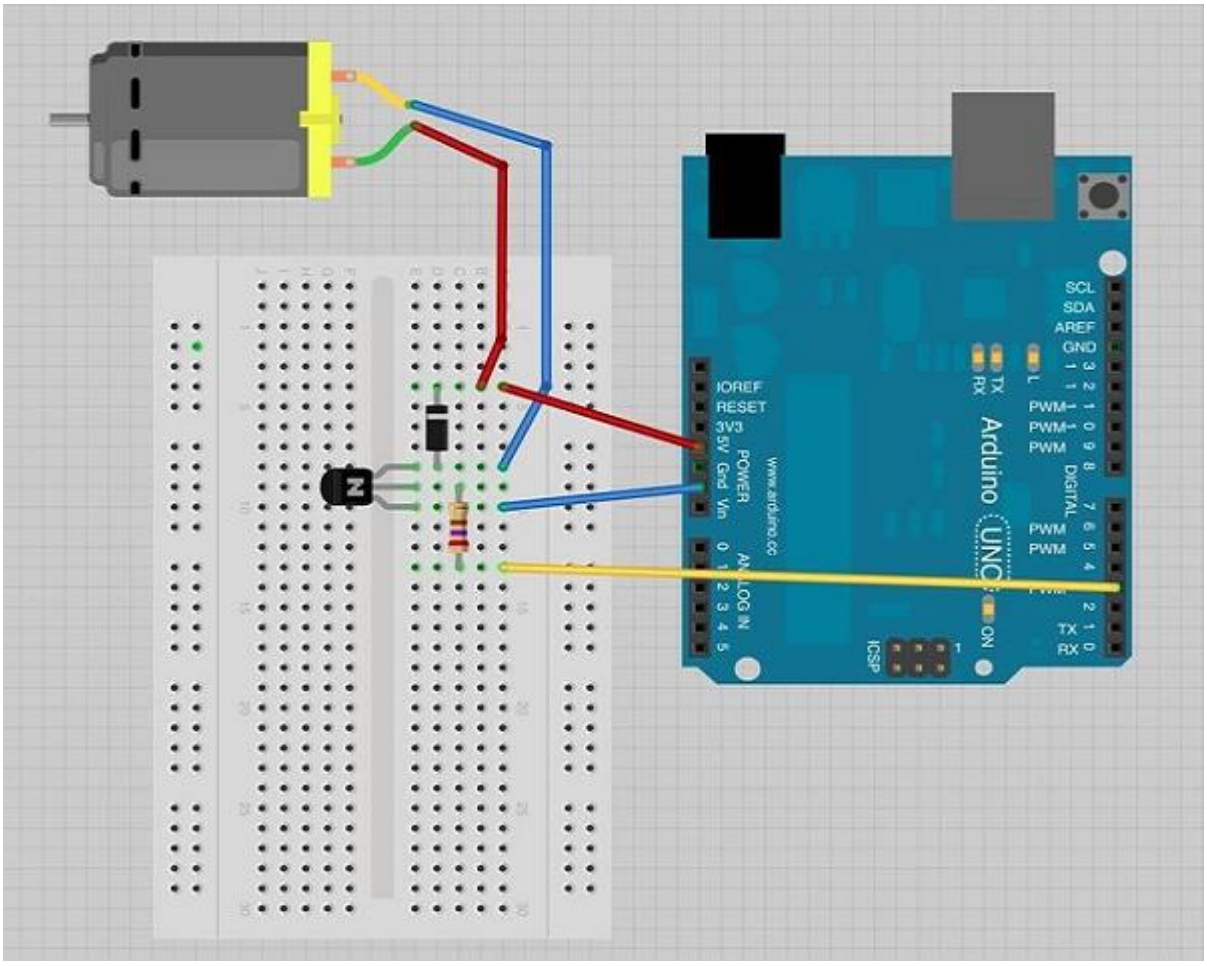


FIGURE 10. MOTOR CONNECTION

Components Required

- 1x Arduino UNO board
- 1x PN2222 Transistor
- 1x Small 6V DC Motor
- 1x 1N4001 diode
- 1x 270 Ω Resistor

Circuit explanation;

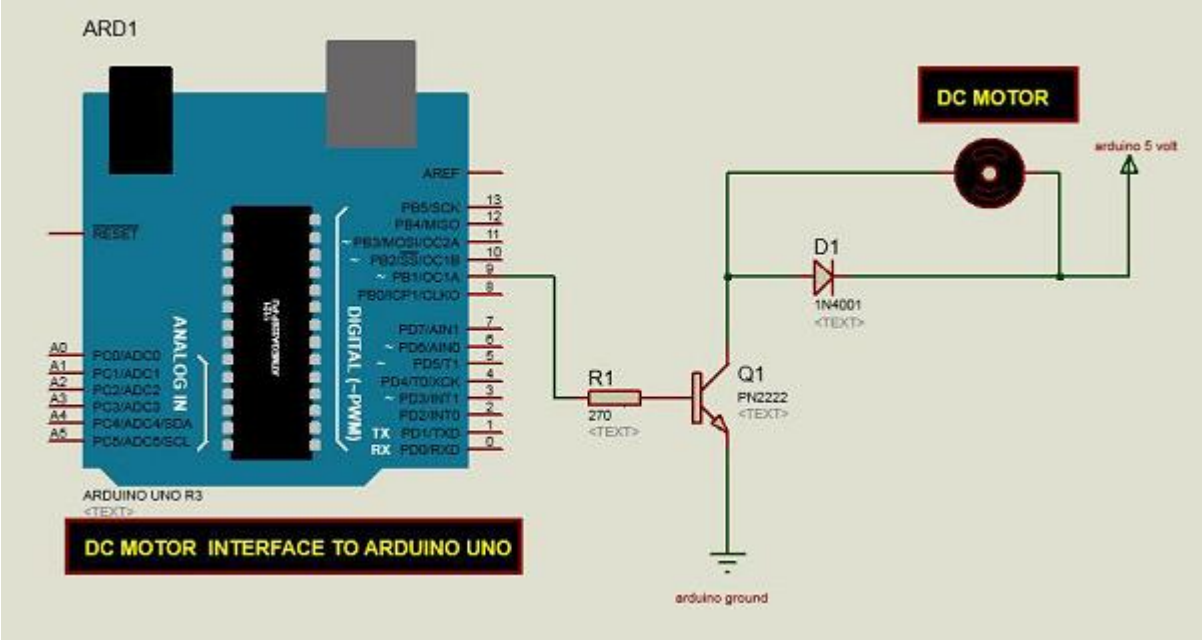


FIGURE 11. MOTOR CIRCUIT

When the circuit is connected as shown above and the vehicle is not overloaded the Arduino board digital pin 9 is (HIGH) that means that current is flowing through it thus going through the base of the transistor via the collector to the diode which is then forward biased and the motor is able to rotate.

When the digital pin is (LOW) no current is able to flow through the base of the transistor. Then in turn the diode is reverse biased due to the current from the 5V supply of the Arduino and the motor circuit is broken therefore the motor cannot rotate.

3.2 Explanation of Arduino code.
HX711 calibration code.

```
/*
  ----------------------------------------------------------------------------
  HX711_ADC
  Arduino library for HX711 24-Bit Analog-to-Digital Converter for Weight Scales
  Olav Kallhovd sept2017
  ----------------------------------------------------------------------------
*/

/*
  This example file shows how to calibrate the load cell and optionally store the calibration
  value in EEPROM

  To implement calibration in your project sketch the simplified procedure is as follow:
     LoadCell.tare();
     //place known mass
     LoadCell.refreshDataSet();
     float newCalibrationValue = LoadCell.getNewCalibration(known_mass);
*/

#include <HX711_ADC.h>
#if defined(ESP8266)|| defined(ESP32) || defined(AVR)
#include <EEPROM.h>
#endif

//pins:
const int HX711_dout = 4; //mcu > HX711 dout pin
```

```
const int HX711_sck = 5; //mcu > HX711 sck pin

//HX711 constructor:
HX711_ADC LoadCell(HX711_dout, HX711_sck);

const int calVal_eepromAdress = 0;
unsigned long t = 0;

void setup() {
  Serial.begin(57600); delay(10);
  Serial.println();
  Serial.println("Starting...");

  LoadCell.begin();
  //LoadCell.setReverseOutput(); //uncomment to turn a negative output value to positive
  unsigned long stabilizingtime = 2000; // preciscion right after power-up can be improved by adding a few seconds of stabilizing time
  boolean _tare = true; //set this to false if you don't want tare to be performed in the next step
  LoadCell.start(stabilizingtime, _tare);
  if (LoadCell.getTareTimeoutFlag() || LoadCell.getSignalTimeoutFlag()) {
    Serial.println("Timeout, check MCU>HX711 wiring and pin designations");
    while (1);
  }
  else {
    LoadCell.setCalFactor(1.0); // user set calibration value (float), initial value 1.0 may be used for this sketch
    Serial.println("Startup is complete");
  }
  while (!LoadCell.update());
```

```
  calibrate(); //start calibration procedure
}

void loop() {
  static boolean newDataReady = 0;
  const int serialPrintInterval = 0; //increase value to slow down serial print activity

  // check for new data/start next conversion:
  if (LoadCell.update()) newDataReady = true;

  // get smoothed value from the dataset:
  if (newDataReady) {
    if (millis() > t + serialPrintInterval) {
      float i = LoadCell.getData();
      Serial.print("Load_cell output val: ");
      Serial.println(i);
      newDataReady = 0;
      t = millis();
    }
  }

  // receive command from serial terminal
  if (Serial.available() > 0) {
    char inByte = Serial.read();
    if (inByte == 't') LoadCell.tareNoDelay(); //tare
    else if (inByte == 'r') calibrate(); //calibrate
    else if (inByte == 'c') changeSavedCalFactor(); //edit calibration value manually
  }
```

```
  // check if last tare operation is complete
  if (LoadCell.getTareStatus() == true) {
    Serial.println("Tare complete");
  }

}

void calibrate() {
  Serial.println("***");
  Serial.println("Start calibration:");
  Serial.println("Place the load cell an a level stable surface.");
  Serial.println("Remove any load applied to the load cell.");
  Serial.println("Send 't' from serial monitor to set the tare offset.");

  boolean _resume = false;
  while (_resume == false) {
    LoadCell.update();
    if (Serial.available() > 0) {
      if (Serial.available() > 0) {
        char inByte = Serial.read();
        if (inByte == 't') LoadCell.tareNoDelay();
      }
    }
    if (LoadCell.getTareStatus() == true) {
      Serial.println("Tare complete");
      _resume = true;
    }
```

```
  }

  Serial.println("Now, place your known mass on the loadcell.");
  Serial.println("Then send the weight of this mass (i.e. 100.0) from serial monitor.");

  float known_mass = 0;
  _resume = false;
  while (_resume == false) {
    LoadCell.update();
    if (Serial.available() > 0) {
      known_mass = Serial.parseFloat();
      if (known_mass != 0) {
        Serial.print("Known mass is: ");
        Serial.println(known_mass);
        _resume = true;
      }
    }
  }

  LoadCell.refreshDataSet(); //refresh the dataset to be sure that the known mass is measured correct
  float newCalibrationValue = LoadCell.getNewCalibration(known_mass); //get the new calibration value

  Serial.print("New calibration value has been set to: ");
  Serial.print(newCalibrationValue);
  Serial.println(", use this as calibration value (calFactor) in your project sketch.");
  Serial.print("Save this value to EEPROM adress ");
  Serial.print(calVal_eepromAdress);
```

```
  Serial.println("? y/n");

  _resume = false;
  while (_resume == false) {
    if (Serial.available() > 0) {
      char inByte = Serial.read();
      if (inByte == 'y') {
#if defined(ESP8266)|| defined(ESP32)
        EEPROM.begin(512);
#endif
        EEPROM.put(calVal_eepromAdress, newCalibrationValue);
#if defined(ESP8266)|| defined(ESP32)
        EEPROM.commit();
#endif
        EEPROM.get(calVal_eepromAdress, newCalibrationValue);
        Serial.print("Value ");
        Serial.print(newCalibrationValue);
        Serial.print(" saved to EEPROM address: ");
        Serial.println(calVal_eepromAdress);
        _resume = true;

      }
      else if (inByte == 'n') {
        Serial.println("Value not saved to EEPROM");
        _resume = true;
      }
    }
  }
```

```
  Serial.println("End calibration");
  Serial.println("***");
  Serial.println("To re-calibrate, send 'r' from serial monitor.");
  Serial.println("For manual edit of the calibration value, send 'c' from serial monitor.");
  Serial.println("***");
}

void changeSavedCalFactor() {
  float oldCalibrationValue = LoadCell.getCalFactor();
  boolean _resume = false;
  Serial.println("***");
  Serial.print("Current value is: ");
  Serial.println(oldCalibrationValue);
  Serial.println("Now, send the new value from serial monitor, i.e. 696.0");
  float newCalibrationValue;
  while (_resume == false) {
    if (Serial.available() > 0) {
      newCalibrationValue = Serial.parseFloat();
      if (newCalibrationValue != 0) {
        Serial.print("New calibration value is: ");
        Serial.println(newCalibrationValue);
        LoadCell.setCalFactor(newCalibrationValue);
        _resume = true;
      }
    }
  }
  _resume = false;
```

```
  Serial.print("Save this value to EEPROM adress ");
  Serial.print(calVal_eepromAdress);
  Serial.println("? y/n");
  while (_resume == false) {
    if (Serial.available() > 0) {
      char inByte = Serial.read();
      if (inByte == 'y') {
#if defined(ESP8266)|| defined(ESP32)
        EEPROM.begin(512);
#endif
        EEPROM.put(calVal_eepromAdress, newCalibrationValue);
#if defined(ESP8266)|| defined(ESP32)
        EEPROM.commit();
#endif
        EEPROM.get(calVal_eepromAdress, newCalibrationValue);
        Serial.print("Value ");
        Serial.print(newCalibrationValue);
        Serial.print(" saved to EEPROM address: ");
        Serial.println(calVal_eepromAdress);
        _resume = true;
      }
      else if (inByte == 'n') {
        Serial.println("Value not saved to EEPROM");
        _resume = true;
      }
    }
  }
  Serial.println("End change calibration value");
```

```
  Serial.println("***");
}
```

vehicle load management code.

```
/*
  ---------------------------------------------------------------------------------
  HX711_ADC
  Arduino library for HX711 24-Bit Analog-to-Digital Converter for Weight Scales
  Olav Kallhovd sept2017
  ---------------------------------------------------------------------------------
*/

#include <HX711_ADC.h>
#if defined(ESP8266)|| defined(ESP32) || defined(AVR)
#include <EEPROM.h>
#endif

#include <Adafruit_FONA.h>

#include <SoftwareSerial.h>
#include "Adafruit_FONA.h"

#define FONA_RX          2
#define FONA_TX          3
#define FONA_RST         4

#define FONA_RI_INTERRUPT  0
SoftwareSerial fonaSS = SoftwareSerial(FONA_TX, FONA_RX);

Adafruit_FONA fona = Adafruit_FONA(FONA_RST);
```

```
char PHONENUM[21] = "+254757164129"; // Enter your Number here.
char welcomemessage[141] = "GSM Kit is working.";
char loadCap_message[141]= "YOUR VEHICLE IS OVERLOADED";

//pins:
const int ledPin = 12;
const int buzzerPin = 13;
const int HX711_dout = 4; //mcu > HX711 dout pin
const int HX711_sck = 5; //mcu > HX711 sck pin
const int Engine = 7;

//HX711 constructor:
HX711_ADC LoadCell(HX711_dout, HX711_sck);

const int calVal_eepromAdress = 0;
unsigned long t = 0;


void setup () {
  Serial.begin(57600); delay(10);
  Serial.println();
   Serial.println(F("Initializing....(May take 3 seconds)"));
  delay(5000);
  fonaSS.begin(9600); // if you're using software serial
  if (! fona.begin(fonaSS)) {           // can also try fona.begin(Serial1)
    Serial.println(F("Couldn't find FONA"));
    while (1);
  }
    Serial.println(F("FONA is OK"));
```

```
  Serial.println("Starting...");


  LoadCell.begin();
  //LoadCell.setReverseOutput(); //uncomment to turn a negative output value to positive
  float calibrationValue; // calibration value (see example file "Calibration.ino")
  calibrationValue = 696.0; // uncomment this if you want to set the calibration value in the sketch
#if defined(ESP8266)|| defined(ESP32)
  //EEPROM.begin(512); // uncomment this if you use ESP8266/ESP32 and want to fetch the calibration value from eeprom
#endif
  EEPROM.get(calVal_eepromAdress, calibrationValue); // uncomment this if you want to fetch the calibration value from eeprom


  unsigned long stabilizingtime = 2000; // preciscion right after power-up can be improved by adding a few seconds of stabilizing time
  boolean _tare = true; //set this to false if you don't want tare to be performed in the next step
  LoadCell.start(stabilizingtime, _tare);
  if (LoadCell.getTareTimeoutFlag()) {
    Serial.println("Timeout, check MCU>HX711 wiring and pin designations");
    while (1);
  }
  else {
    LoadCell.setCalFactor(calibrationValue); // set calibration value (float)
    Serial.println("Startup is complete");
  }


  pinMode(ledPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(Engine, OUTPUT);
}
```

```
void loop() {

  static boolean newDataReady = 0;
  const int serialPrintInterval = 0; //increase value to slow down serial print activity

  // check for new data/start next conversion:
  if (LoadCell.update()) newDataReady = true;
// get smoothed value from the dataset:
  if (newDataReady) {
    if (millis() > t + serialPrintInterval) {
      float i = LoadCell.getData();
      if (i >= 400.0) {

        tone(buzzerPin, 100);
        digitalWrite(ledPin, HIGH);
        delay(200);

        noTone(buzzerPin);
        digitalWrite(ledPin, LOW);
        delay(200);

        digitalWrite(Engine,LOW);
        fona.sendSMS(PHONENUM,loadCap_message);
        delay(200);

        Serial.println("----------- ENGINE STOPPED-----------");
        Serial.println(i);
  }
    else {
```

```
      noTone(buzzerPin);
      digitalWrite(ledPin, LOW);
      digitalWrite(Engine, HIGH);

      Serial.println("ENGINE STRARTED");
      Serial.println(i);
    }

  }
 }


}
```

Explanation

Arduino programming language can be divided in three main parts: functions, values (variables and constants), and structure. Functions: For controlling the Arduino board and performing computations. Variables; this are Arduino data types and constants. Structure; The elements of Arduino (C++) code, for example sketch that includes the loop () and setup ().

The above code has function and library calls, variable and constants declarations and the main code. The main code consists of void setup () and void loop ().

Void setup ()

This is where the setup code is; the setup code enables the loop code to know where it will gather its information from, and it will only run once. This part mainly checks if the major components are working that is; Arduino board, HX711 and the GSM module. It also assists the loop code to get the calibrated data.

Void loop ()

In void loop(), your code will repeat over and over again. In this part of the code, the if else statement was used, the if else statement will keep repeating itself in order for it to gather information constantly and ensure the system is always running. Therefor if the load capacity of the vehicle exceeds the stated limit the system will stop the engine and the alert system will kick in. Else if the vehicle limit is not exceeded the motor will always be on.

3.3 Circuits or block diagrams.

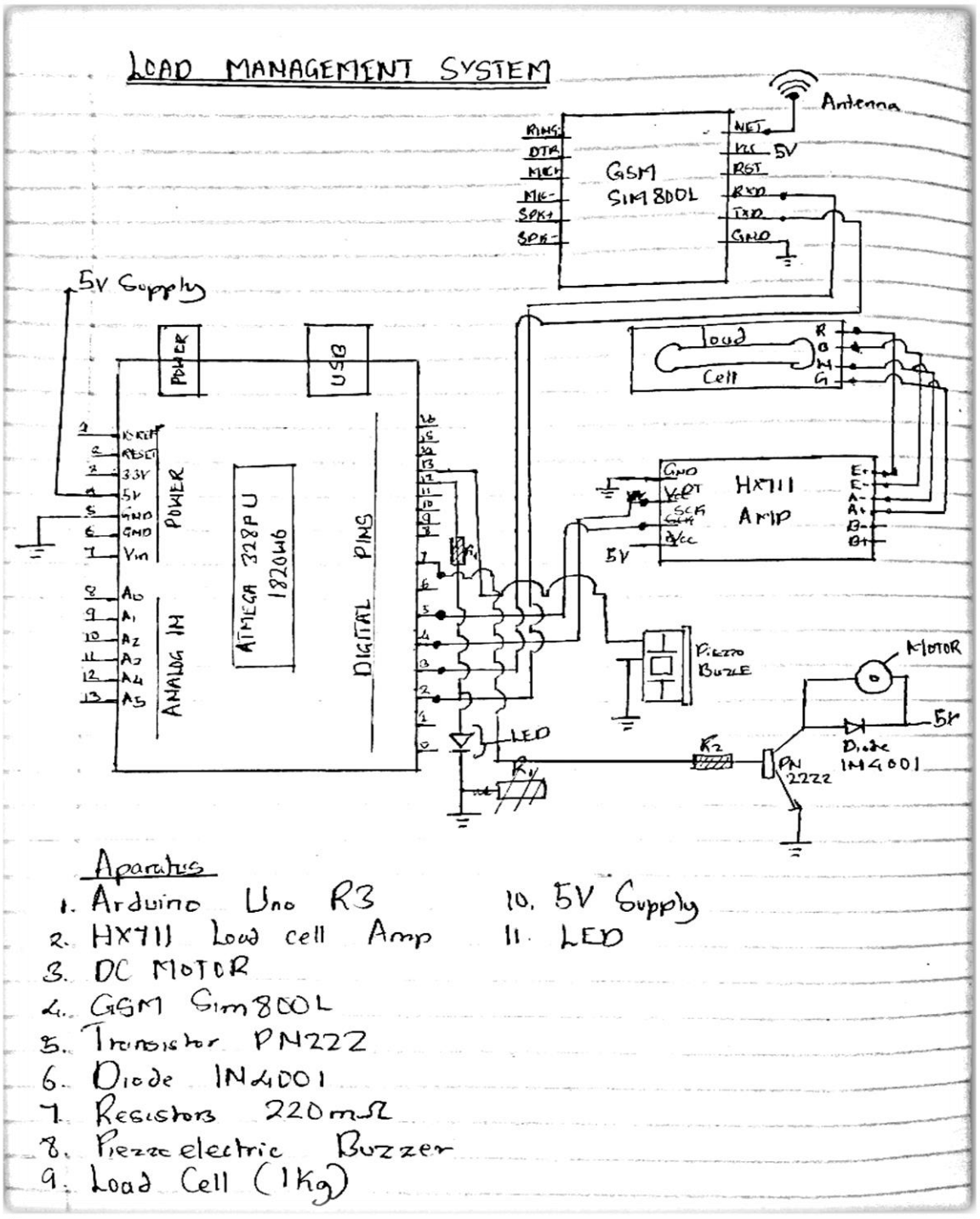


FIGURE 12. WHOLE CIRCUIT DIAGRAM

3.4 Explanation of how the project was carried out.

Procedure.

1) The circuit is connected as shown in figure 12.
2) Then the Arduino board is connected to the computer via a USB cable.
3) Then the calibration code is uploaded to the Arduino board via Arduino IDE.
4) Then open the serial monitor and follow the steps as directed.
5) After the load cell is calibrated the load cell is able to measure the weight given within its limit.
6) After the calibration the rest of the code is uploaded and the system was able to run accordingly.

CHAPTER FOUR
RESULTS
4.0 Results for a system that can weigh vehicle load weight.

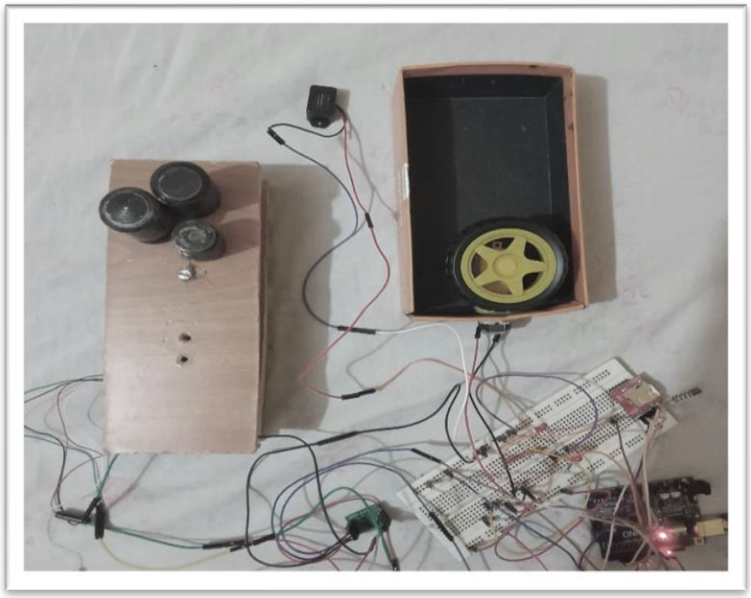


FIGURE 13. Results for load weight measurement.

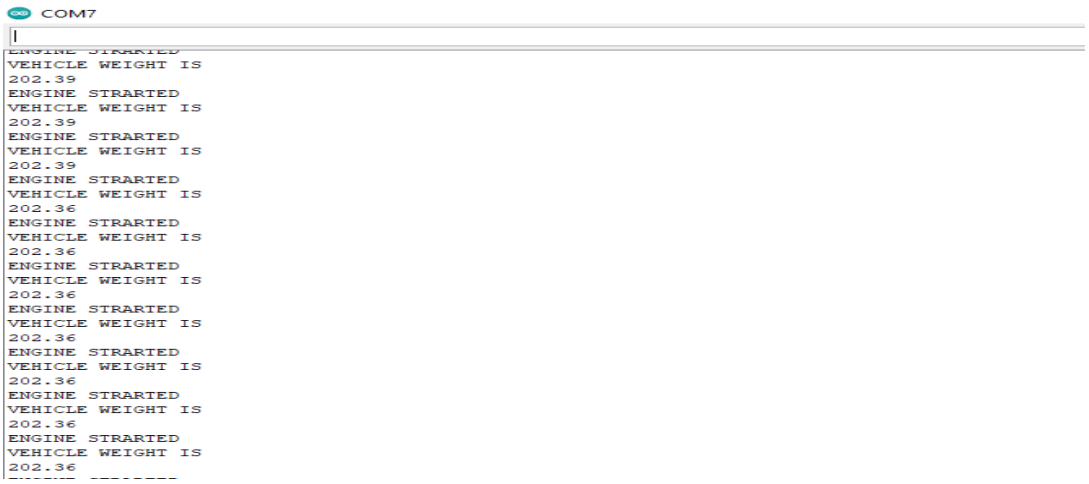


FIGURE 14. Load measurement values.

figure 13 and 14 show the whole circuit and the weight values read from the load cell in grams respectively, Since the capacity of the load cell is limited to one kilogram.

4.2 Results for a system that can send a message alert if the vehicle is overloaded.

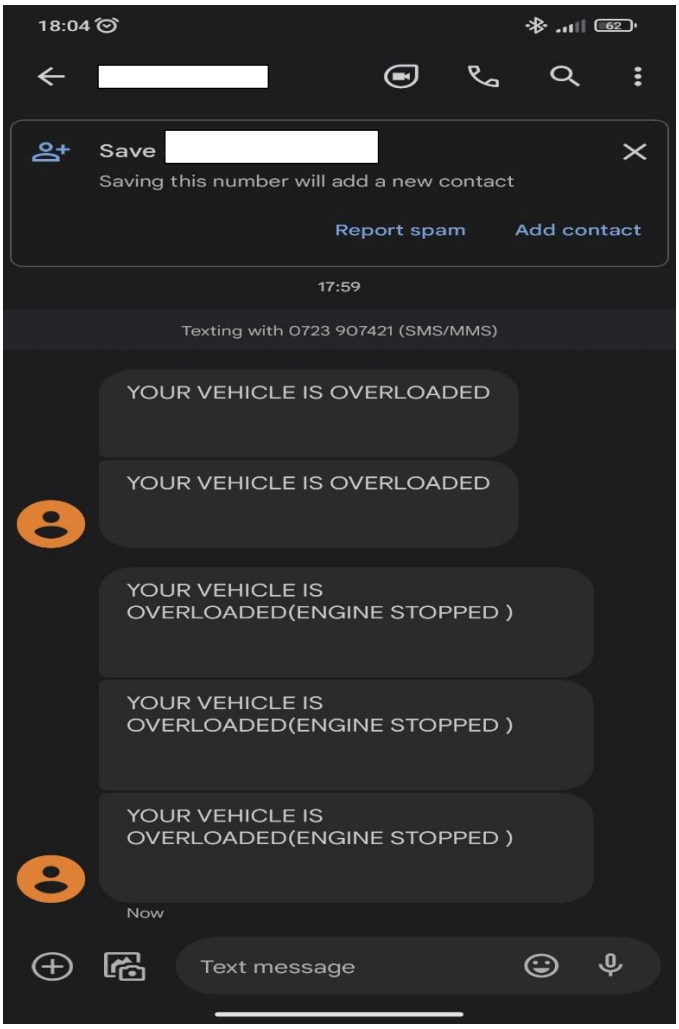
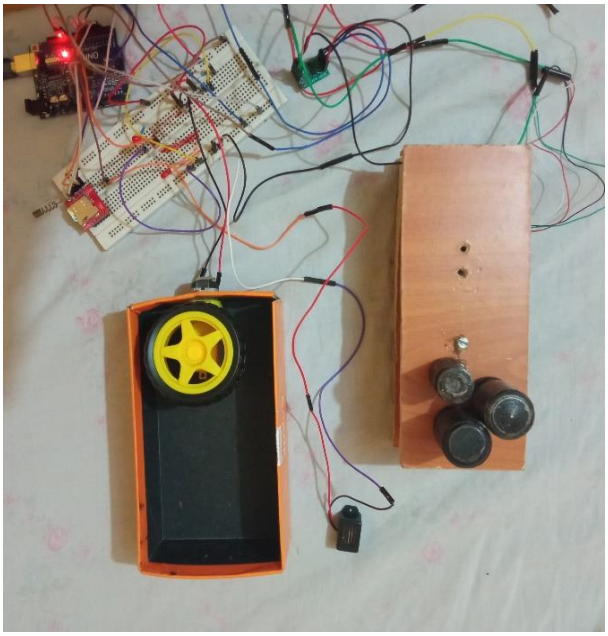
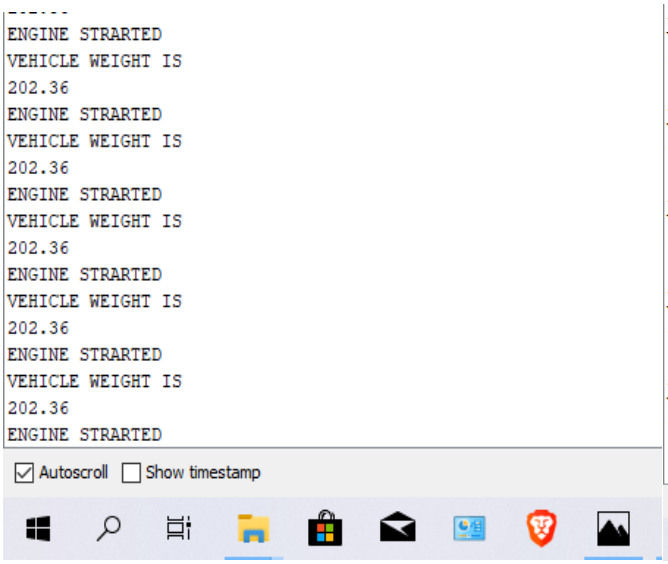


FIGURE 15. Results for SMS alert.

The above image shows the massage received from the GSM module, to inform the authorities that this vehicle is overloaded and the engine has been stopped.

4.3 Results for a system that can stop the engine if the vehicle is overloaded.





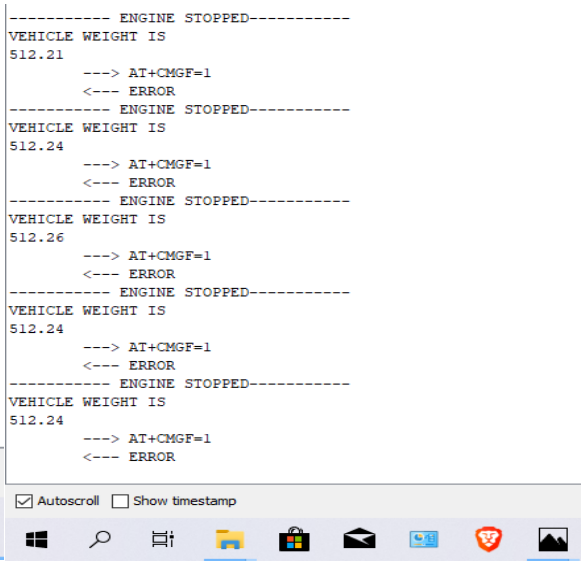


FIGURE 16. Not Overloaded

FIGURE 17. Overloaded and engine stopped

Figure 16 simulates when the vehicle is empty or not yet overloaded, it shows that the engine/motor is on. While figure 17 shows that the vehicle is overloaded and the engine/motor has stopped.

References

Anyango, G.J. (2011). Addressing the persistent delays at the weighbridges. Stakeholders Workshop Addressing Bottlenecks at the Port of Mombasa and the Weighbridges. Nyali International Beach Hotel, Mombasa.

Benekohal, R. F. , El-Zohairy, Y. , Forrler, E. , and Aycin, M.(1998) . Delay and Truck Traffic Characteristics for Evaluation of AVI and WIM for CVO at Williamsville Weigh Station. Illinois Department of Transportation, Department of Civil Engineering, University of Illinois at Urbana-Champaign.

Cunagin, W, W. A. Mickler, Charles Wright. January 1, 1997 . Evasion of Weight-Enforcement Stations by Trucks. Florida Department of Transportation. Center for Transportation Studies, University of North Texas, Denton.

Chan, Y.C. (2008). Truck Overloading Study in Developing Countries and Strategies to minimize its impacts. Masters Thesis, Southeast University, China.

East African Community (2012). EAC Vehicle Load Control Bill, 2012. East African Secretariat, Arusha.

East African Online Transport Agency (2012). The Northern Corridor Transport Network. Retrieved September 11th, 2013 from http://www.eaotransport.com/.

European Union (2006). Best Options Study "Assessing the Current ALC in Kenya" Recommendations. Final Report. Scott International.

https://www.pololu.com/product/2191#:~:text=The%20Arduino%20Uno%20R3%20is,to%2Duse%20Arduino%20computer%20program.

https://predictabledesigns.com/the-sim800-cellular-module-and-arduino-a-powerful-iot-combo/

https://www.instructables.com/How-to-Interface-HX711-Balance-Module-With-Load-Ce/

https://instrumentationtools.com/load-cell-working-principle/#:~:text=Load%20cell%20is%20a%20sensor,different%20kinds%20of%20load%20cells.

https://en.wikipedia.org/wiki/Piezoelectricity.

https://en.wikipedia.org/wiki/Piezoelectric_speaker

https://www.tutorialspoint.com/arduino/arduino_dc_motor.htm

https://www.electronicdesign.com/technologies/components/article/21796017/understanding-led-application-theory-and-practice

https://en.wikipedia.org/wiki/Light-emitting_diode

https://repository.up.ac.za/handle/2263/7924

https://iopscience.iop.org/article/10.1088/1757-899X/694/1/012003/meta

http://erepository.uonbi.ac.ke/handle/11295/52363

http://41.89.49.13:8080/xmlui/handle/123456789/1286

http://erepository.uonbi.ac.ke/handle/11295/98330

JICA and PADECO (2011). Study for the Harmonization of Vehicle Overload Control in the East African Community, Final Report.AFD; Publ. 11-007.

Ministry of Roads (2011). Road Sector Investment Programme (2010-2014) Report. Kenya Government Press.

Ministry of Transport, Kenya. (2009). Integrated National Transport Policy. The National Transport Policy Committee. Nairobi; pp 57-69.

Masyhur, A H et al 2019 IOP Conf. Ser.: Mater. Sci. Eng. 694 012003.

UNIVERSITY OF CONNECTICUT. Downloaded on May 19,2021 at 02:17:21 UTC from IEEE Xplore.

Workshop Addressing Bottlenecks at the Port of Mombasa and the Weighbridges.Nyali International Beach Hotel, Mombasa.

Workshop Addressing Bottlenecks at the Port of Mombasa and the Weighbridges. Nyali International Beach Hotel, Mombasa.